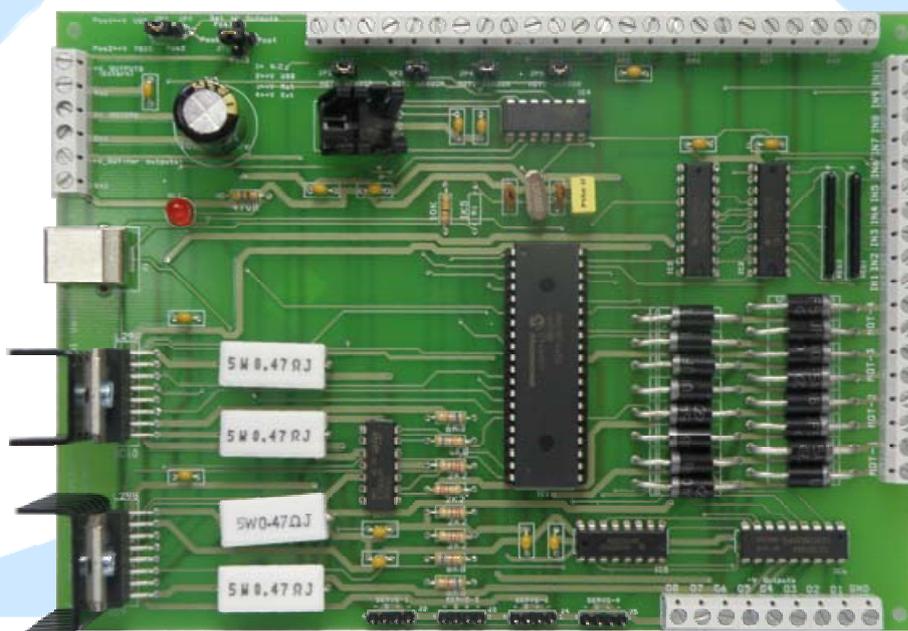




Opencockpits



locardUSB DCMotors Plus Manual

Index:

IOCARDUSB DCMOTORS PLUS MANUAL.....	1
INDEX:.....	2
INTRODUCTION:.....	3
USB DCMOTORS PLUS:.....	3
SPECIFICATIONS:.....	3
SCHEME AND MEASURES:.....	4
NOTAS SOBRE LA ELECTRÓNICA :.....	4
DESCRIPTION OF THE CONNECTORS:.....	5
CONFIGURATION OF JUMPERS:.....	6
CONNECTIONS:.....	7
STARTING UP THE CARD:.....	8
DEFINITION OF THE ELEMENTS OF THE CARD FOR ITS USE WITH SIOC.....	8
<i>Programation notes:</i>	9
TEST ASSEMBLY:.....	10
<i>Definition in Sioc.ini.</i>	11
<i>Definition of the test script.</i>	12
LOADING THE SCRIPT AND TESTING:.....	12
TESTING THE USBDCMOTORSPLUS:.....	13
LINKS OF INTEREST:.....	14

Introduction:

The USBDCMotorsPlus has been designed to be able to interact with different motors, inputs and outputs. It can be used for the automation of control of gases, controlled force (force feedback) and different types of automation systems. This card is connected through the USB port to the computer and is managed through the Protocol IOCP using at least version 4.2 of SIOC.

USB DCMotors Plus:

Unlike her younger sister, USB DCMotors, this card allows the use of much more powerful motors, including the possibility of also using servo-motors, as well as the use of both analog and digital inputs and outputs.

It also adds to the firmware an auto positioning system in order to manage automated self-assessment systems without any programming.

The card also includes auto-sensing of the motors. I.e., through analog inputs the card can measure at all times the intensity that passes through the coils of motors with what the user can know what effort is being made by the engines. This is very useful to detect the end of run in the movements allowing us to act depending on the strength that is generated in the motors.



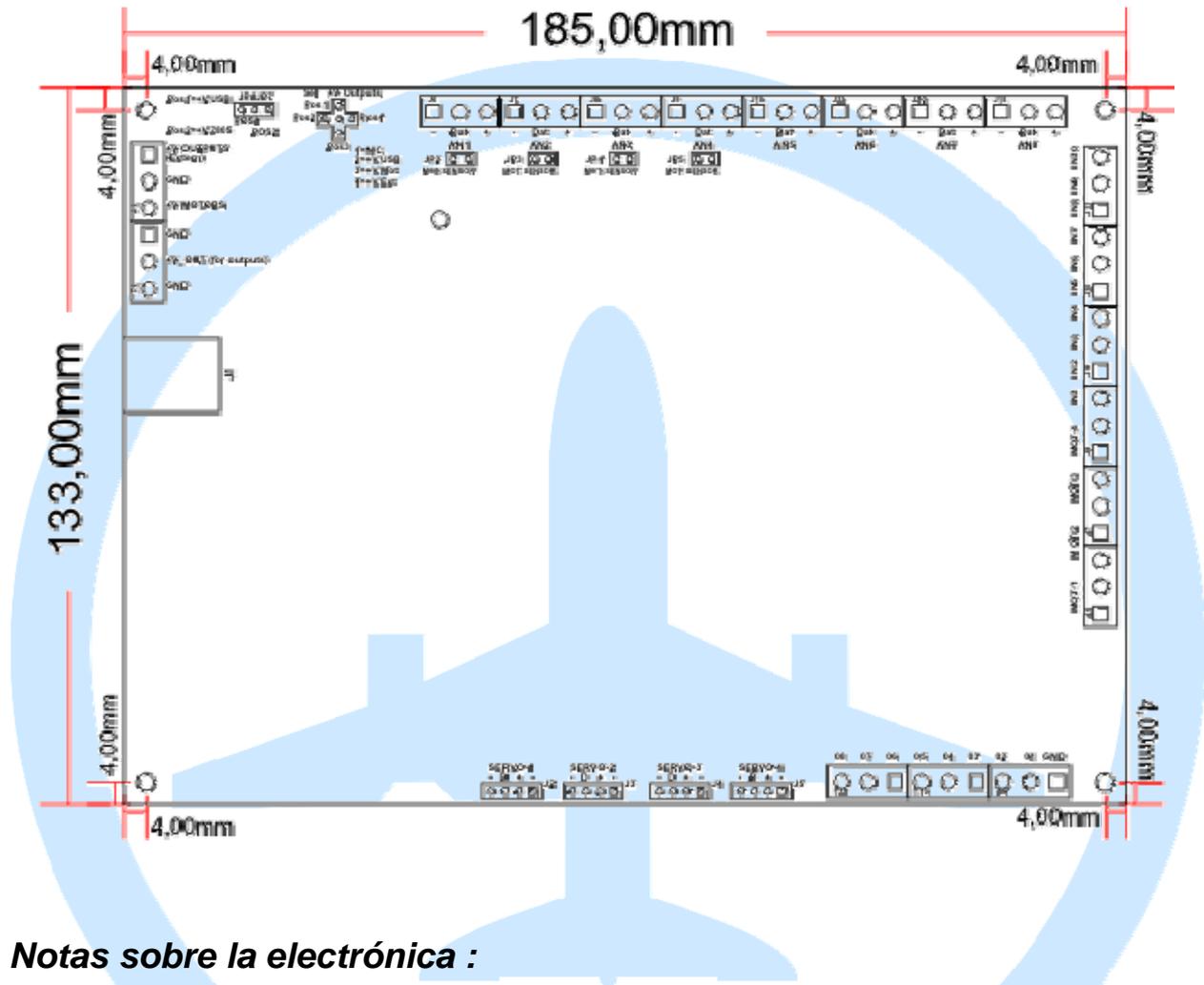
In the future, without changing electronics, only with a different firmware, you may also manage high power stepper motors.

Specifications:

- Input current for motors up to 48 volts.
- Up to 4 DC motors. 2 constant amps, 2.5 Amp non-constant, 3 amp peaks.
- Motors (8-bit) PWM speed control.
- Direction control of the motors.
- Over-temperature protection.
- Up to 4 servo-motors type RC with resolution 10bits control.
- 8 Digital outputs of up to 50 V/500mA
- Selectable 5V internal power supply for use with servo motors or digital outputs.
- Second power supply optional for digital outputs.
- 8 Analog inputs of 10-bit internal / 8-bit external.
- Ability to auto-sensing power of each motor via analog inputs.
- Motors self-positioning system (max. 3 motors) through the use of analog inputs (using potentiometers, magnetic sensors, etc.).
- Up to 10 digital inputs.

Scheme and measures:

- J1 = USB CONNECTOR.
- JP1, JP6 JP7 = 3 PIN CONNECTORS.
- JP2...JP5 = 2 PIN CONNECTORS.
- J2 (Servo1)...J5 (Servo4) = 4 PIN CONNECTORS.
- J6 (W-1)...J13 (An8), T1...T11 = SCREW CONNECTORS.



Notas sobre la electrónica :

Depending on the motors and the use that is given to them, the card can overheat, needing therefore good ventilation and for long working periods, the use of a fan would be advisable.

Some DC motors can generate a good deal of electrical noise. To avoid this we recommend the use of "debugging" capacitors and even if a large amount of electronic fluctuations is detected in the inputs, it is advisable to use of electrolytic capacitors in analog input connectors (between GND and positive about 10mF, 63V or higher) to avoid problems reading the analog inputs, and even if the electrical noise peaks are large, avoid the automatic switch-off of the card from the USB port on the computer.



The card does not have protection device against short circuits and overloads, it only has protection against overheating, so it would be advisable to use some kind of protection (fuse or overvoltage protector).

Description of the connectors:

- J1 = USB connector, allows direct connection to your computer, at the time of connecting the computer will recognize the card and will install the device.
- J2...J5 = connectors for the servo-motors. The connector is 4, to connect 3 to the left or to the right 3.
Pins 2, 3 and 4 are used for the servos and Opencockpits gauges and compatible (2 red / +, 4/black - and yellow 3/data).
Pins 1, 2 and 3 are FUTABA, JR, HITEC servos and compatible. Anyway, it is highly recommended to check the cables:

COLOR	VALUE
Black/ Brown	GND
White/ yellow/orange	DATA
Red or blue	+

TABLA CONEXIONES POR MARCAS		
FABRICANTE	ORIGINAL	USB SERVO
FUTABA J		
HITEC		
JR		
AIRTRONICS "Z"		

- J6...J13 = analog inputs AN1 screw connectors...AN8:

GND.

Data.

+ 5V.

AN5, AN6 and AN7 inputs can be used for the self-positioning of MOT-1, MOT-2 and MOT-3 motors.

- T1 and T2 = screw connector:

+ V outputs (pin 1 and pin 5), positive external power to outputs. Connect a GND to negative and positive to one of the pins. The other will be used as common to the exits, because in this case the common outlet is the positive and not the negative.

GND (pins 2, 4 & 6) = negative for power sources.

+ V motors (pin 3) = positive power to engines. The negative can be connected to any GND.

- T3, T4 and T5 = screw connectors:

MOT-1...MOT-4, connection of the poles of motors 1 to 4.

- T5...T8 = screw connectors for:

IN1...IN10, analogue inputs. If they are open (not connected) a 0 is returned in the variable of software. If they connect to GND, a 1 value will be returned to the corresponding variable.

- T9, T10 and T11 = screw connectors:

GND, extra connection to GND for entries.

O1...O8, digital outputs. In these terminals the negatives of each output are connected. It is important to note that the common output is the positive (+ V_OUT). If a 1 is placed in the corresponding software variable, current will pass between the positive outputs and each output terminal.

Configuration of jumpers:

JP1-JP7:

Position 1, the card takes 5 volts from the USB port. This is useful for testing or when servomotors will be used and only a small supply is needed.

Position 2, 5 volt current is taken from the internal power, which in turn is generated from the power of the engines.

Jumper switch + V for outputs:

Pos 1, unused position.

Pos 2, the output will use the 5 volt power (which is selected by jumpers JP1-JP7).

Pos 3, the output will use the power of the engines.

Pos 4, the output will use a different external power supply.

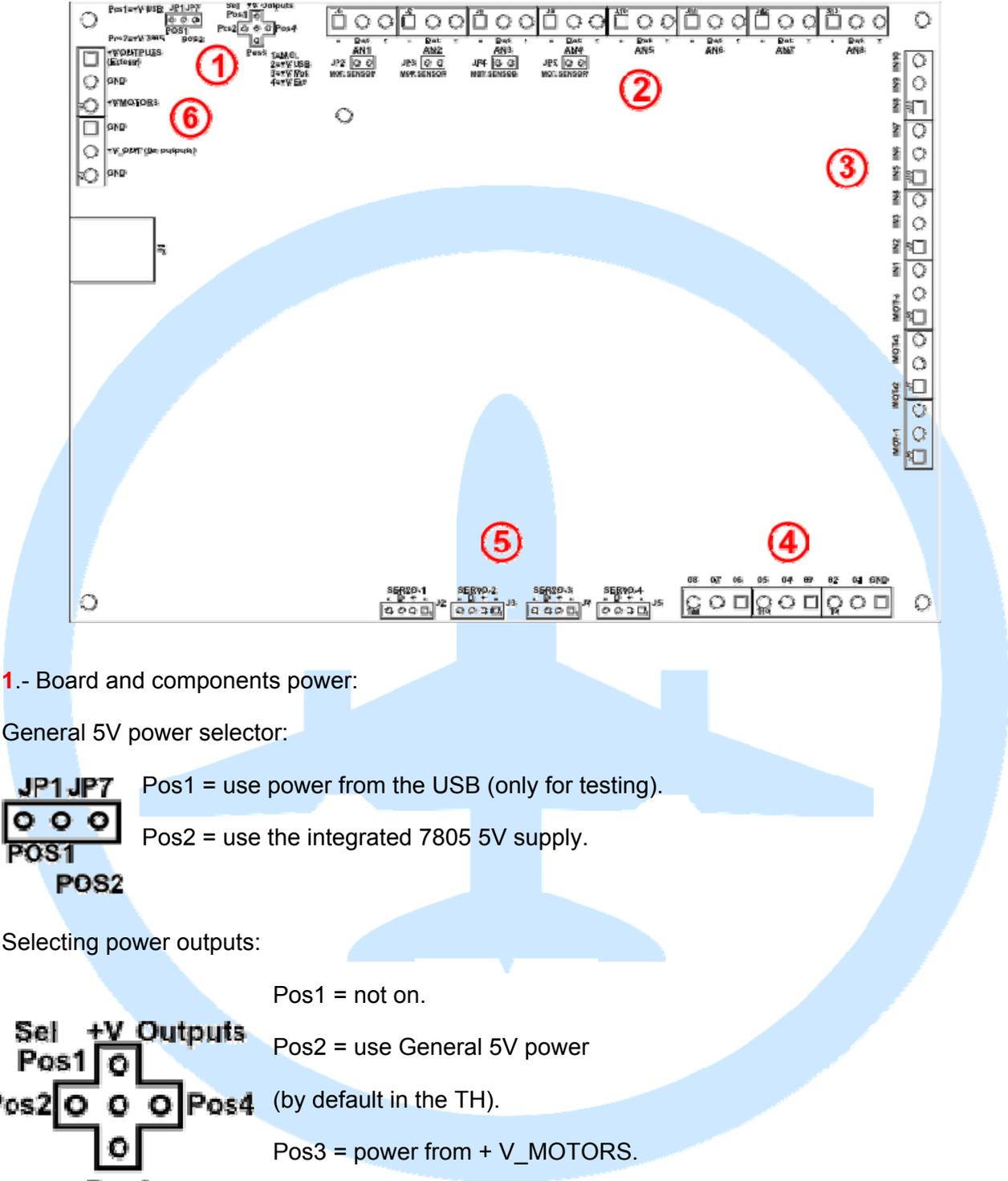
JP2 to JP5 connectors:

Open, entries AN1 to AN4 work as normal analog inputs.

Closed, entries AN1 to AN4 are connected to the sense of MOT-1, MOT-2, MOT-3 and MOT-4 motors respectively.

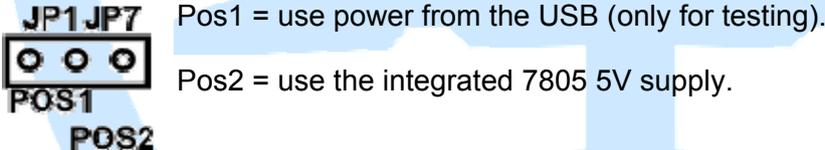
Connections:

For a better explanation of the connections, an image is attached:

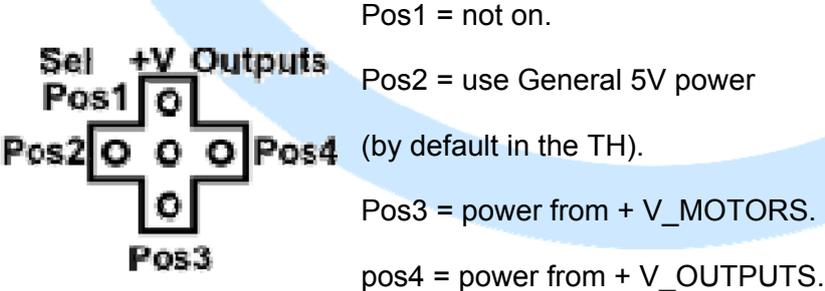


1.- Board and components power:

General 5V power selector:



Selecting power outputs:



2.- 8 analog inputs.

The first 4 with closed jumper measure the current in motors 1 to 4 (sense).

The following 3 with activated software are used for auto positioning of Motors 1 to 3.

3.- Digital inputs 1 to 10, are connected by shorting any GND.

4.- Digital outputs. Negative is connected to these connectors and positive to the general connector V_OUT (for outputs) into the power strip.

5.- Servo connections. Double connector prepared for the two most common types, pay attention when connecting the servos.

PINS (1, 2 & 3): Futaba, Hitec, JR, etc.	PINS (2, 3 & 4): Opencockpits servos and gauges.
	

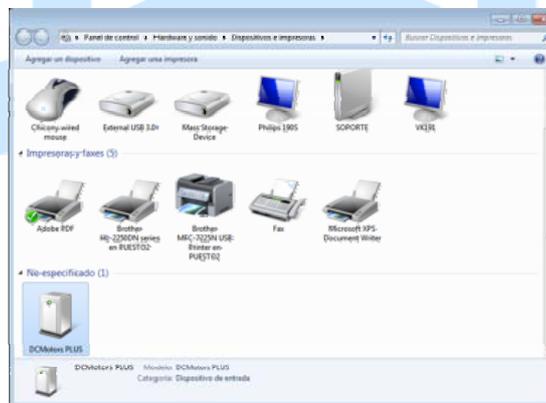
6.- + V_OUTPUTS: To connect if we want specific power outputs (do not use on the Throttle).

+ V_MOTORS: General motors and internal sources (7805) power.

+ V_OUT: (output with the same power than + V_OUTPUTS), common positive power for outputs.

Starting up the card:

We already know the IOCard USB DCMotors Plus board, next step is to connect it to the computer, test it and test its operation. Once connected to your computer, it will be immediately recognized as an USB device.



Important notice: Never connect or disconnect the motors with the card connected to the USB port and power.

Definition of the elements of the card for its use with Sioc.

Programming and configuration of the card is via the SIOC program version 4.2 or higher and is tested by IOCPConsole.

The definition of all the elements of the card for its use with Sioc are:

```
// *****
// * Definición de USB DCMotors Plus - By Manuel Velez - www.opencockpits.com
// *****

Var 0001, name Servo1, Link IOCARD_SERVO, device X, Output 1, PosL 0, PosC 512, PosR 1023
...
Var 0004, name Servo4, Link IOCARD_SERVO, device X, Output 4, PosL 0, PosC 512, PosR 1023
Var 0005, name Motor1, Link IOCARD_MOTOR, device X, Output 0, Aceleration 0
...
Var 0008, name Motor4, Link IOCARD_MOTOR, device X, Output 3, Aceleration 0
Var 0009, name output1, Link IOCARD_OUT, device X, Output 0
...
Var 0016, name output8, Link IOCARD_OUT, device X, Output 7
Var 0017, name autopos1A, Link IOCARD_CONFIG, device X, Output 1
Var 0018, name autopos1B, Link IOCARD_CONFIG, device X, Output 2
Var 0019, name autopos2A, Link IOCARD_CONFIG, device X, Output 3
Var 0020, name autopos2B, Link IOCARD_CONFIG, device X, Output 4
Var 0021, name autopos3A, Link IOCARD_CONFIG, device X, Output 5
Var 0022, name autopos3B, Link IOCARD_CONFIG, device X, Output 6
Var 0023, name input1, Link IOCARD_SW, device X, Input 0
...
Var 0032, name input10, Link IOCARD_SW, device X, Input 10
Var 0033, name analog1, Link IOCARD_ANALOGIC, device X, Input 1, PosL 0, PosC 128, PosR 255
...
Var 0040, name analog8, Link IOCARD_ANALOGIC, device X, Input 8, PosL 0, PosC 128, PosR 255
```

Programation notes:

Entries are activated with 1 and deactivated with 0.

Outputs are activated with 1 and deactivated with 0.

Analog inputs range from 0 to 255.

Servo-motors range from 0 to 1023 (in the 0 position the servo-motor runs out of power and has free movement: the servo can be displaced manually or with a motor but with care to avoid damaging the gears).

Motors have two forms of operation:

1. By default or if each motor A and B configuration variables are set to 0. The motor is software controlled using the motor variable declared in the script, so that with a value of 0 to 127 (stop 0, 127 maximum speed) will go in one direction, with a value of 128 to 255 (stopped 128), 255 maximum speed) will go in the other direction.

Range of Values	Speed
0	Stopped
1→127	min→max (one-way)
128	Stopped
129→255	min→max (opposite direction)

2. If A or B configuration variables have a some data, the corresponding motor is set in self-positioning mode. Thus motors are controlled by the card, trying to accelerate and turn to carry the value of the respective analog input, to the target value that is selected in the variable of the motor. Upon reaching point the motor will stop automatically. Configuration variable A defines the acceleration that the engine will start and stop with, the variable B is the sensitivity adjustment to the arrival area. Sometimes we will have to play with these values until you find the desired operation.

Test Assembly:

We will use a potentiometer, a DC motor, that we can obtain from any appliance such as a printer or a toy, a led, a switch, and a servo.

We connect our elements in the following manner:

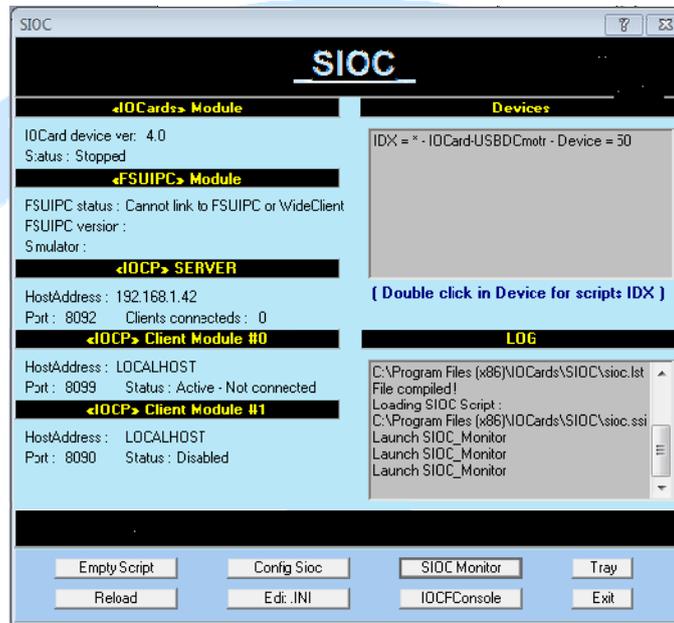


Our test engine and servo operate at 5V, so we will feed 5V to the USB-DCMotors Plus card from an external power supply for the tests, i.e. closed JP1 in POS2 and JP6 in POS3. The motor will be connected to MOT-1 pins 1 and 2 output (the direction of rotation can be changed either via SIOC, either via hardware changing the connection of the motor cables). The potentiometer is connected to input AN1 (J6 with open JP2), the led to the output O1 and to the

common positive, the switch to IN1 and the servo to Servo-1 (in position 2-3-4 because it is a servo configured with Opencockpits wiring).

Definition in Sioc.ini.

Start SIOC (version 4.2 or higher), observe the information given in the upper right window and see if our card is properly configured. If it is, will appear with a number of recognized IDX (if it is not properly declared a * symbol will appear), with its name and the USB port (Device = XXX) where it is connected.



If it is not properly configured, let's do this:

Click the EDIT.INI button, the text editor will open with our configuration sioc.ini file, then look for the [IOCard Master] section, in it we will see how to declare a DCMotors Plus USB:

```
[MASTER =(Indice_device), (type),(numero_de_tarjetas),(numero_device)]
[type = 14: USBDCmotorPLUS card]
```

That is:

```
MASTER =YY,14, 1,XX
[USB Plus DCMotors]
```

YY = number to identify the card in our scripts (IDX in Sioc = DEVICE in script).

14 = card type, in this case a USB Plus DCMotors.

1 = number of USB DCMotors Plus cards. This value is fixed and cannot be changed.

XX = number of the USB port where the card is connected.

We recommend to use the same number of IDX as used to declare what type of card it is, as long as we only have a single card of the same type connected, in the event that they are more, the number of IDX set is irrelevant, but cannot be repeated. In our case, we shall declare the DCMotors Plus USB as:

```
MASTER=14,14,1,271
[USB DCMotors Plus]
```

271 = USB port on our computer evidence in your case number 20th of your screen of Sioc.

Save and exit, press the RELOAD button and our card will already be correctly configured and able to be managed with a script.

It is important to take a good look to the assigned IDX, which in our example is 14, in order to configure and use the card, the potentiometers and the motors coupled to it. Now let's create the test script.

Definition of the test script.

For this chapter, we will rely on the definition of the elements of the card. We will only use at the moment the definitions of the elements that we are actually going to try, to avoid doing this manual excessively heavy, all other items are equal.

Create a script named Prueba_USB_DCMotors_Plus.txt with a simple text editor such as notepad:

```
// *****
// ** Definition of USB DCMotors Plus - By Manuel Velez - www.opencockpits.com
// *****
// * FileName : Prueba_USB_DCMotors_Plus.txt
// * Date : 12/07/2012
Var 0001, name Servo1, Link IOCARD_SERVO, device 14, Output 1, PosL 0, PosC 512, PosR 1023
// This is the SERVO-1
Var 0002, name Motor1, Link IOCARD_MOTOR, device 14, Output 0, Aceleration 0
// This is the MOT-1
Var 0003, name output1, Link IOCARD_OUT, device 14, Output 0
// This is the O1
Var 0004, name input1, Link IOCARD_SW, device 14, Input 0
// This is the en-1
Var 0005, name analog1, Link IOCARD_ANALOGIC, device 14, Input 1, PosL 0, PosC 128, PosR 255
// This analogue input is the W-1
// END OF FILE
```

We kept it and proceed to upload it to Sioc, with this simple script can test the features of the USB DCMotors Plus.

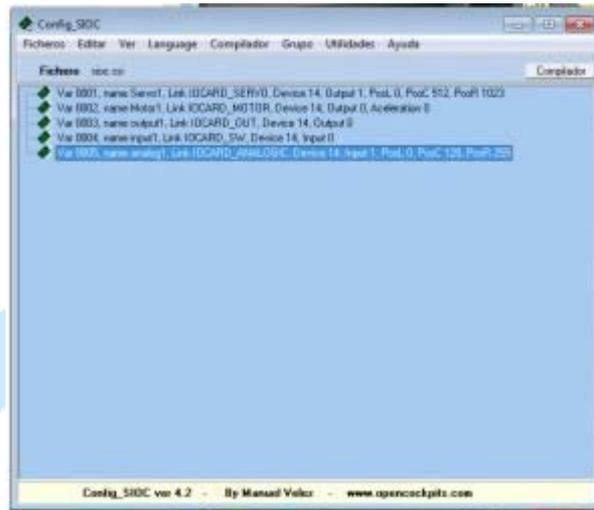
Loading the script and testing:

In this section we are going to load the test script that we just made in SIOC, we start SIOC, click the Edit.INI button, sioc.ini will appear in the text editor, look for the section "[** SIOC **]" and inside, the subsection "[configuration file] / [Configuration File]" then replace the existing line by the following:

```
CONFIG_FILE=.\Prueba_USB_DCMotors_Plus.txt
```

Save the changes and close the text editor. Make sure that the script that we did in the previous chapter and we referenced in sioc.ini is in the same directory as Sioc.exe.

As we still have Sioc running, first click the Reload button to update changes made to sioc.ini, and then click Config Sioc, which opens another window and this window will contain the script that we have declared in sioc.ini, if does not contain errors.

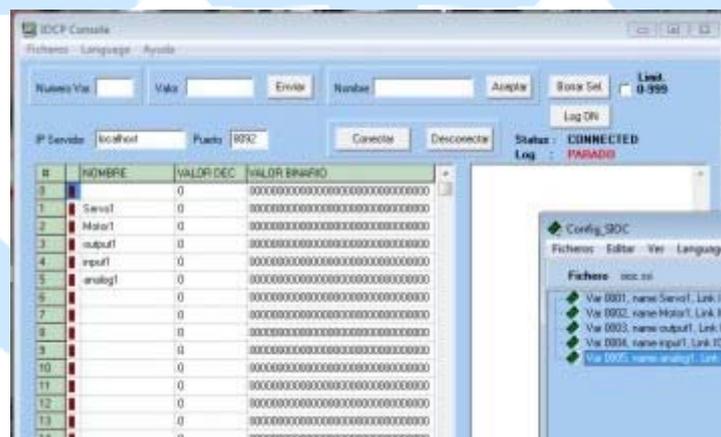


If there are no errors close Config Sioc and will continue with the DCMotors Plus USB testing, if there are errors, will tell us which has been found and we will have to review the script to fix them.

If all has gone well, connect the items to the card and Cunén it externally (although the burden is going to get is small but don't want to overheat the components of the plate) and then connect the card to the computer which, together with Sioc, will recognize it and will be ready to start testing.

Testing the USBDCMotorsPlus:

Start Sioc, press the IOCPConsole button and the following window appears:



Click Log ON and connect, the variables will be the actual values at that time:

Servo1: Here we can put the position that we want the servo to take, if you type in the value position 200, the servo will go to that position, if we introduce the 800 position it will go to that position, etc. If you enter a value of 0, the servo will be unlocked and you can move freely with your hands (with great care not to break the physical limits of the gears).

Motor 1: Displays the value 0 (stopped), if we send a value between 1 and 127 it will rotate in one sense, if we send the value 128 or 0 it will stop again, and if we send a value between 129 and 255 it will rotate in the opposite direction. The motor speed will increase as you increase the sent value (remember the table on page 10).

Output1: Displays the value 0 initially and if we send the value 1, the led connected in O1 will turn on and the led will go off with a 0. (Remember that leds need a resistor to avoid burning and O outputs or have a common positive).

Input1: Displays the current status of the switch that is connected to IN1, 0 if the switch is open and 1 if it is closed.

Analog1: Displays the reading of the potentiometer connected to AN1, if we turn the potentiometer shaft, displayed values will vary in unison.

The application of the card to use with simulators is exactly the same that with other IOCards, that is why we invite you to read the manuals of other Opencockpits items and the SIOC software manuals. Thanks for trusting us.

That ends this manual.

Links of interest:

Support Zone for customers:

<http://www.opencockpits.com/catalog/info/>

